

産業用データベース・ミドルウェアの開発

1 W-5 — 実時間データ獲得・提供システム RTDS —

井戸 譲治† 島川 博光† 竹垣 盛一†

†三菱電機(株)産業システム研究所

1 まえがき

産業用システムでは、外界データの実時間収集・管理機構は必要不可欠である。一般的な DBMS を用いた場合、データ管理の面で高い機能を提供できる反面、実時間性に問題があり、産業用システムの性質を考慮したアーキテクチャが必要となる。本稿では、産業用データベースシステムに求められるデータ収集機構の持つ特質と、これに特化したデータベースシステムである RTDS の実現について述べる。

2 RTDS の概要

2.1 プラントデータ処理

プラントシステムにおけるデータ処理は、現在ないしは最近のデータを用いてのプラント操業のための処理と、長期に渡るデータを用いての各種解析をもとにした品質管理や経営管理のための処理に大別される。前者では、複雑なデータ構造や大量データの処理、排他制御といった機能性よりも実時間性が重視される傾向にあり、後者はその逆である。また、これらのデータ利用処理の前提となるデータ収集処理は、さらに厳しい実時間性を要求される。

また、プラントでは、様々な製造機器が結合されて複雑な構造を持つのに加え、時間とともに物品(製品)がそれらの上を移動して行くため、製品データと機器データを関連付けることができなければならない。移動する物品の扱いは単純な関係モデルでは困難であり、それに適した他のモデルが必要となる。

2.2 位置付けと機能

RTDS は、移動する物品の関連付けのような複雑なデータ処理などは RTVS[2] に任せ、関連付けの対象となる時系列データの収集管理と、RTVS の能動機構のトリガとなる状態変化(イベント)の通知を担当する。即ち、機能性よりも実時間性に重きを置いている。

実時間データ収集 一定周期ないしはプラント側からの通知をトリガとして、データの収集を行う。あらゆるデータ処理の前提となるデータ収集処理は、一度取

りこぼすと二度とそのデータを得られないことから、厳しい実時間性が要求される。

時系列データの提供 獲得されたある時刻のスナップショットデータを時刻順に並べた時系列データを提供する。異なる周期で獲得されたデータを並べて見せるために、時刻印を結合キーとした join を行う。データの提供は、検索要求に受動的に反応するだけでなく、一定周期や、後述する状態変化検出をトリガとして能動的に行うことができる。

状態変化の検出・通知 新たに収集された現在のデータ、及びそれ以前に獲得された比較的最近のデータに対して、あらかじめ条件を設定しておくことで、データ収集時に条件検査を行い、成立した場合にこれを上位に通知することができる。また、この通知を自分自身に発行することで、これをトリガとしたデータ収集や提供を行うことができる。

3 セマンティクスを活用した実現

3.1 時系列データ処理の特性

プラントの状態を表す時系列データの収集管理処理には、以下のような特徴がある。

1. データの入力は一定周期ないしは何らかのイベントに対応して機械的に行われ、収集されたデータの変更や削除は通常行われない。
2. 検索時に指定される属性が時刻属性に偏る。他の属性を指定したとしても、多くの場合は、時刻属性値(時系列の期間)の指定との併用である。
3. 処理のアドラインは一様でなく、データ収集が厳しい実時間性を要求されるのに対し、データ提供は比較的緩やかである。またデータ提供処理の中でも最近のデータを対象としたものと過去のデータを対象としたものでは、その用いられ方から、後者は時間制約を持たないとしてよいことが多い。

RTDS では、これらの特性を最大限に活用した構成をとる [3]。

3.2 記憶構造と排他制御

記憶構造 データを格納するバッファはリング状に構成され、収集順に格納される。リングバッファに格納されたデータは上書きされる前に、同様にリング状に

構成されたファイルに退避される。収集された時系列データに対する削除/修正を認めていないため、リングバッファ及びファイル上で、時系列データは時刻順に並ぶことが保証される。このような記憶構造により、

- 最近のデータは主記憶上にあるため高速アクセスが可能
- そのままバイナリサーチによる検索が可能
- 時刻属性に基づく結合 (join) 操作が比較的効率良く実現できる
- 平均値や微積分値といった、一定期間の一連の場面列へのアクセスが容易

といった利点が生まれる。

排他制御 新たに収集されたデータの格納は、常にリング構造の最古位置に対して行われる。そこで、リング構造の最新/最古位置を保持した管理情報へのアクセスのみを優先度継承セマフォを用いて排他制御し、リング構造上のデータそのものへはロックせずにアクセス可能とした。万一読み出し途中のデータを書き潰された場合のために、楽観的読み出しを行なう。

ロックの対象となる管理情報は小さなものであり、読み書きに必要な時間は無視できるほど小さい。また、優先度継承セマフォを用いているため、優先度逆転による不当なブロックは起こらない。従って共有データへのアクセスで生じるはずのブロック時間は無視できるほど小さくなる。そのため、各タスクの最悪実行時間が見積もれ、静的なスケジューリング判定が可能となる。

また、時間制約を持たない読みだし処理のディスクアクセスのために、実時間性が要求される収集データのディスクへの格納が不当にブロックされることのないよう、アクセスサイズの最適化ならびにディスクの占有制御を行っている。

3.3 時系列の間合せと応答

上述の時系列データ処理の特性から、基準時刻、場面データ (スナップショット) 間の時間的間隔 (以後レートと呼ぶ) とその数、許容誤差および取り出すべき変数の組 (スキーマ) のみを検索パラメータとして指定可能とした。

スキーマ s , 基準時刻 b_0 , レート τ , 許容誤差 a , 基準時刻より前のレート数 n_{old} , 基準時刻以降のレート数 n_{new} , 場面補正モード m (raw/aligned/reformed), に対し、次のように検索を行なう。

```

start ← b0 - τ · nold
for n ← 0 to (nold + nnew - 1)
do
    b(n) ← start + τ · n
    foreach si ∈ {s1, ..., sk | s = s1 ⋈ ... ⋈ sk}
    do
        si(n) ← siの時系列の中で
            区間 (b(n) - a, b(n)) 内で
            最大のタイムスタンプをもつ場面
    done
    s1(n) ⋈ ... ⋈ sk(n) をモード m に従って計算
    結果を時間順に書き出す
done
    
```

$s_1(n) \bowtie \dots \bowtie s_k(n)$ の計算はモード m により、以下のように異なる (図 1)。

raw 通常の outer join. タイムスタンプが異なれば異なる場面となるため、空値が多く含まれる。

aligned 各 $s_i(n)$ のタイムスタンプを $b(n)$ に補正した上で outer join を行なう。

reformed どの s_i も存在しない区間 $(b - a, b]$ に対しても、タイムスタンプに b を持ちすべての変数値が空値である場面 (空場面と呼ぶ) を挿入する。

通常 join 演算は重い処理であるが、前述のように時系列データに特化した記憶構造を持つため、比較的効率良く join を実行することができる。

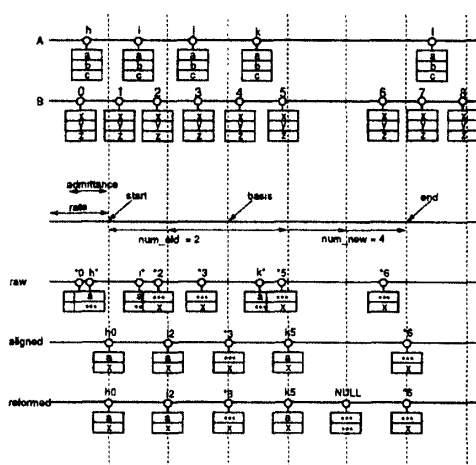


図 1: 時系列の検索

4 あとがき

本稿では、プラントデータ獲得・提供ミドルウェア RTDS について述べた。RTDS はトンネル監視制御システムなどに適用されている。さらに、各プラントへの適用を容易にするためのビルダ/ジェネレータの開発も行っている。

参考文献

- [1] H.Shimakawa, H.Ohnishi, I.Mizunuma, M.Takegaki: Acquisition of Temporal Data for Real-Time Plant Monitoring, Proc. of RTSS'98 (1993)
- [2] 高田 秀志, 島川 博光, 竹垣 盛一: 産業用データベース・ミドルウェアの開発-時間/場所/物体モデルとトラッキング機構-, 情報処理学会第 55 回全国大会予稿集 1W-06 (予定) (1997)
- [3] 井戸 譲治, 島川 博光, 竹垣 盛一: 実時間データ獲得システム RTDS の評価, 電子情報通信学会研究報告 DE96 - 24 ~ 53, pp.79-84 (1996)